The Distribution of First Digits

In this lab, you will explore the distribution of first digits in real data. For example, the first digits of the numbers 52, 30.8, and 0.07 are 5, 3, and 7 respectively. In this lab, you will investigate the question: how frequently does each digit 1-9 appear as the first digit of the number?

Question 0

Make a prediction.

1. Approximately what percentage of the values do you think will have a *first* digit of 1? What percentage of the values do you think will have a first digit of 9?
2. Approximately what percentage of the values do you think will have a *last* digit of 1? What percentage of the values do you think will have a last digit of 9?

(Don't worry about being wrong. You will earn full credit for any justified answer.)

ENTER YOUR WRITTEN EXPLANATION HERE.

1. Approximately 70% of the time 1 will appear as first digit because it is the first number to be recursed over. The opposite for 9, I think the probability that the first digit will be 9 I think is 30%.
2. I dont think there is much of a difference why 1 will be appear more or less than 9 as the last digit. I think they both have about the same probabilty so my guess is that for both the probability will be around 15% since all digits from 1 to 9 have same chance of being the last digit.

Question 1

The S&P 500 (https://en.wikipedia.org/wiki/S%26P_500_Index) is a stock index based on the market capitalizations of large companies that are publicly traded on the NYSE or NASDAQ. The CSV file `sp500.csv` contains data from February 1, 2018 about the stocks that comprise the S&P 500. We will investigate the first digit distributions of the variables in this data set.

Read in the S&P 500 data. What is the unit of observation in this data set? Is there a variable that is natural to use as the index? If so, set that variable to be the index. Once you are done, display the `DataFrame`.

In [1]:
```
# ENTER YOUR CODE HERE.
import pandas as pd
df = pd.read_csv("sp500.csv")
df.head()

df_by_name = df.set_index("Name")
print(df)
df_by_name.head()
```

```
           date  Name      open     close     volume
0    2018-02-01   AAL   $54.00   $53.88    3623078
1    2018-02-01  AAPL  $167.16  $167.78   47230787
2    2018-02-01   AAP  $116.24  $117.29     760629
3    2018-02-01  ABBV  $112.24  $116.34    9943452
4    2018-02-01   ABC   $97.74   $99.29    2786798
..          ...   ...      ...      ...        ...
500  2018-02-01   XYL   $72.50   $74.84    1817612
501  2018-02-01   YUM   $84.24   $83.98    1685275
502  2018-02-01   ZBH  $126.35  $128.19    1756300
503  2018-02-01  ZION   $53.79   $54.98    3542047
504  2018-02-01   ZTS   $76.84   $77.82    2982259

[505 rows x 5 columns]
```

Out[1]:

Name	date	open	close	volume
AAL	2018-02-01	$54.00	$53.88	3623078
AAPL	2018-02-01	$167.16	$167.78	47230787
AAP	2018-02-01	$116.24	$117.29	760629
ABBV	2018-02-01	$112.24	$116.34	9943452
ABC	2018-02-01	$97.74	$99.29	2786798

ENTER YOUR WRITTEN EXPLANATION HERE.

1 as the first digit appears the most than other digits.

Question 2

We will start by looking at the `volume` column. This variable tells us how many shares were traded on that date.

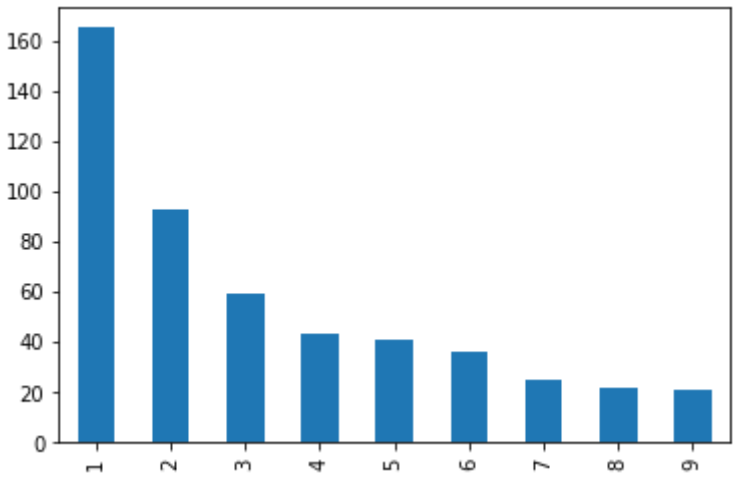
Extract the first digit of every value in this column. (*Hint:* First, turn the numbers into strings. Then, use the text processing functionalities (https://pandas.pydata.org/pandas-docs/stable/text.html) of `pandas` to extract the first character of each string.) Make an appropriate visualization to display the distribution of the first digits. (*Hint:* Think carefully about whether the variable you are plotting is quantitative or categorical.)

How does this compare with what you predicted in Question 0?

In [2]:
```
# ENTER YOUR CODE HERE.
df.volume = df.volume.apply(str)
first_digits = df.volume.str[0]
print (first_digits.value_counts())


import matplotlib
%matplotlib inline
first_digits.value_counts().plot.bar()
```

```
1    165
2     93
3     59
4     43
5     41
6     36
7     25
8     22
9     21
Name: volume, dtype: int64
```

Out[2]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fde3bfcda90>
```



ENTER YOUR WRITTEN EXPLANATION HERE.

It turns out 1 does appear very frequently as the first digit. My guess turned out to be correct for both cases 1 and 9.

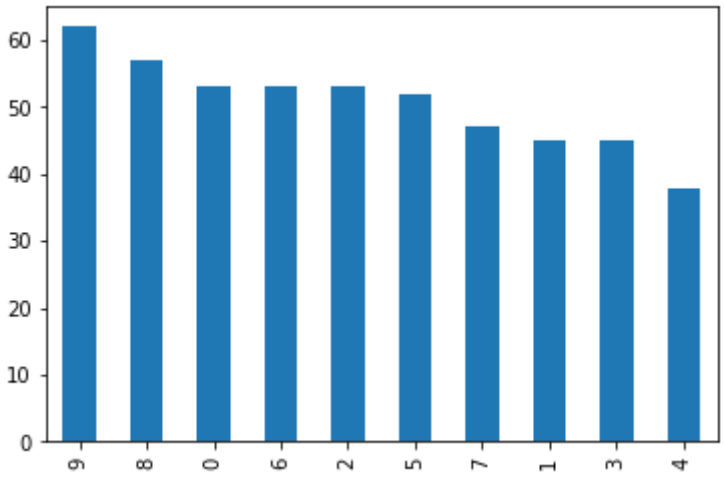
Question 3

Now, repeat Question 2, but for the distribution of *last* digits. Again, make an appropriate visualization and compare with your prediction in Question 0.

In [3]:
```
# ENTER YOUR CODE HERE.
df.close = df.close.apply(str)
last_digits = df.close.str[-1]
print (last_digits.value_counts())


import matplotlib
%matplotlib inline
last_digits.value_counts().plot.bar()
```

```
9    62
8    57
0    53
6    53
2    53
5    52
7    47
1    45
3    45
4    38
Name: close, dtype: int64
```

Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x7fde39c714e0>



ENTER YOUR WRITTEN EXPLANATION HERE. the frequency in which 1 or 9 appear as last digits is the same in the data, my theory was wrong.

Question 4

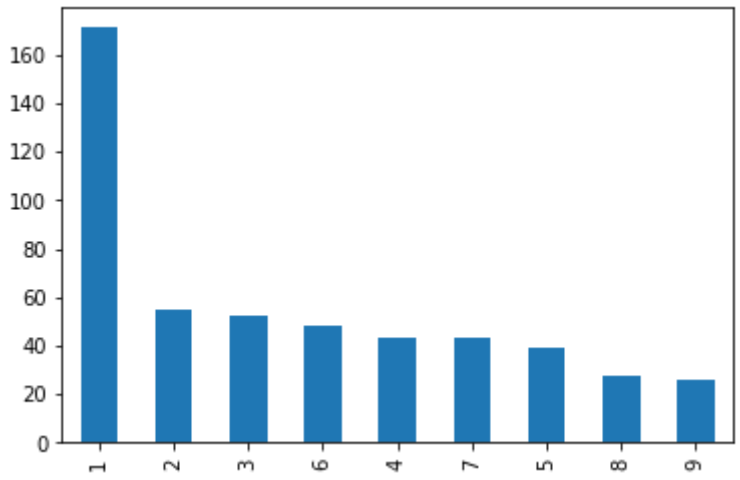
Maybe the `volume` column was just a fluke. Let's see if the first digit distribution holds up when we look at a very different variable: the closing price of the stock. Make a visualization of the first digit distribution of the closing price (the `close` column of the `DataFrame`). Comment on what you see.

(*Hint:* What type did `pandas` infer this variable as and why? You will have to first clean the values using the text processing functionalities (https://pandas.pydata.org/pandas-docs/stable/text.html) of `pandas` and then convert this variable to a quantitative variable

In [4]:
```
# show histogram of elements in array
import numpy
import pylab
import matplotlib

df.close = df.close.apply(str)
first_digits = df.close.str[1]
print ("first_digits:",first_digits.value_counts())



%matplotlib inline
first_digits.value_counts().plot.bar()
```

```
first_digits: 1    171
2     55
3     52
6     48
4     43
7     43
5     39
8     28
9     26
Name: close, dtype: int64
```

Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x7fde3951c6a0>



ENTER YOUR WRITTEN EXPLANATION HERE.

Submission Instructions

Once you are finished, follow these steps:

1. Restart the kernel and re-run this notebook from beginning to end by going to `Kernel > Restart Kernel and Run All Cells`.
2. If this process stops halfway through, that means there was an error. Correct the error and repeat Step 1 until the notebook runs from beginning to end.
3. Double check that there is a number next to each code cell and that these numbers are in order.

Then, submit your lab as follows:

1. Go to `File > Export Notebook As > PDF`.
2. Double check that the entire notebook, from beginning to end, is in this PDF file. (If the notebook is cut off, try first exporting the notebook to HTML and printing to PDF.)
3. Upload the PDF to iLearn.
4. Have the TA check your lab to obtain credit.